

Proposed Comal Standard

Proposal for PRINT USING

Purpose

To allow the user to specify a mask for output to format numerical and string data.

Syntax

PRINT USING <format string> : <expression list> [; | ,]

The format string is evaluated and its value must be in the form

<format mask> { <format mask> }

<format mask> ::= <numeric mask> | <string mask> | <character>

<character> ::= Any character
(characters with special meanings in format statements must be preceded by an ^ to produce literal output)

<string mask> ::= '<' { '<' } | '>' { '>' } | '=' { '=' }

<numeric mask> ::= [<sign specifier>] [<numeric field>]
[. <numeric field>] [<exponential field>]
| '/' { '/' }

<exponential field> ::= 'E' [<sign specifier>] <numeric field>

<sign specifier> ::= '+' | '-'

<numeric field> ::= '#' { '#' } | '@' { '@' }

Semantics

The print using statement regulates output by using a string as a mask. For each character in the mask there must be a one to one correspondence with the output on the screen (except for characters proceeded by a ^). The variable list specifies the values to be output in the mask. If there are too many or too few values an error is generated. (It could be that in the event of too few values the mask for the missing values would be printed.)

String masks are in 3 different forms:-

<

Left justify the string and put spaces in the remaining positions in the mask.

>

Right justify the string and put spaces in the remaining positions in the mask.

=

Centre the string in the mask and put spaces in the remaining positions in the mask. If the string will not centre exactly the string should be printed to the left.

In the event of the string being too large to fit into the mask, the mask characters should be printed.

Numeric masks are in 2 forms:-

Right justified

#

A digit should be placed at this position unless it would be a leading zero in which case a space is used. If no sign specifier is used then a minus sign can be printed in front of the leftmost digit.

@

A digit should be placed at this position including leading zeros. If no sign specifier is used then the minus sign is output in the leftmost character position in the mask.

.

The decimal point is forced into this position number. The numeric field after the decimal point will be filled to the field length available rounded to the correct length.

E

This specifies the start of the exponential field and forces scientific notation. This also forces normalisation of the number output.

Left justified

/

This specifies left justification of a number. There are no other modifiers in this format.

Right and left justified

+

A sign specifier forcing the printing of the sign of the number.

-

A sign specifier which is replaced by a minus sign if the number is negative and by a space if the number is positive.

If any other character is used inside the PRINT USING statement then it is output. Any of the control characters should be preceded by a ^.

Examples

PRINT USING "##.### \$ ###.## " : 37.6687 , -7

will output

"37.669 \$ -7.00 "

PRINT USING "<<<< ^> =====" : "test" , "test"

will output

"test > test "

PRINT USING "+//////// value -####.####E+##" : 27.5 , 31.41

will output

"+27.500 value 0003.1410E+01"

B\$ = Str\$(a, " " ")